**#Installing required R libraries**

install.packages("tidyverse")

library(readr)

install.packages("forecast")

library(forecast)

install.packages("fpp2")

library(fpp2)

library(TTR)

library(dplyr)

library(ggplot2)

library(readxl)

**#Reading data from excel file**

CESC\_MMYYYY\_data\_edit <-read\_excel("C:/Users/rumic/Desktop/Desktop/CESC/Cesc\_data/CESC\_MMYYYY\_data\_edit.xlsx",

col\_types = c("date", "numeric", "text"))

View(CESC\_MMYYYY\_data\_edit)

cesc\_mp = CESC\_MMYYYY\_data\_edit

**#Checking if import of data is correct**

glimpse(cesc\_mp)

**#Converting dataset into timeseries**

cesc\_mp=ts(cesc\_mp[,2])

**#Splitting the data into Training and Test**

cesc\_mp\_train = subset(cesc\_mp, Class == 'Train')

cesc\_mp\_test = subset(cesc\_mp, Class == 'Test')

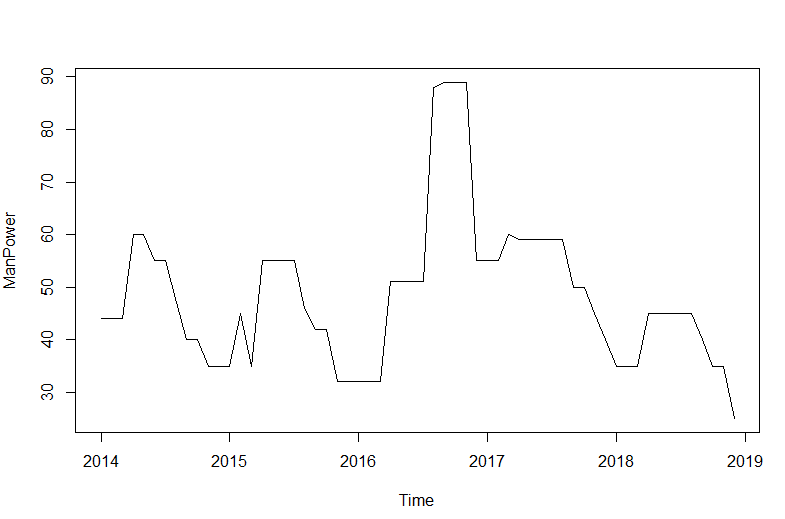
nrow(cesc\_mp\_train);nrow(cesc\_mp\_test)

cesc\_mp\_ts = ts(cesc\_mp\_train[,2], start=c(2014,1), end = c(2018,12), frequency = 12)

cesc\_mp\_ts\_test = ts(cesc\_mp\_test[,2], start=c(2019,1), end = c(2019,12), frequency = 12)

**#Plotting original time series**

plot(cesc\_mp\_ts)



**#Creating a function to calculate MAPE errors of models**

mape = function(actual, pred){mape =

mean(abs((actual - pred)/actual))\*100

return (mape)}

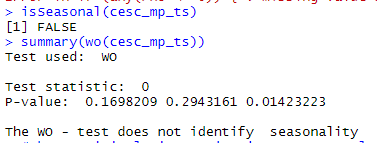
**#Seasonality test**

install.packages("seastests")

library(seastests)

isSeasonal(cesc\_mp\_ts)

summary(wo(cesc\_mp\_ts))



**#Thus, original time series is not seasonal**

**#Stationarity test:**

**#Null hypothesis: Time series is not stationary**

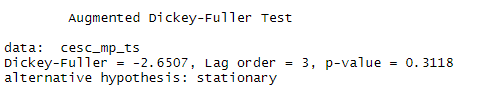
**#Alternative: Time series is stationary**

**#ADF test for stationarity**

install.packages("tseries")

library(tseries)

adf.test(cesc\_mp\_ts)

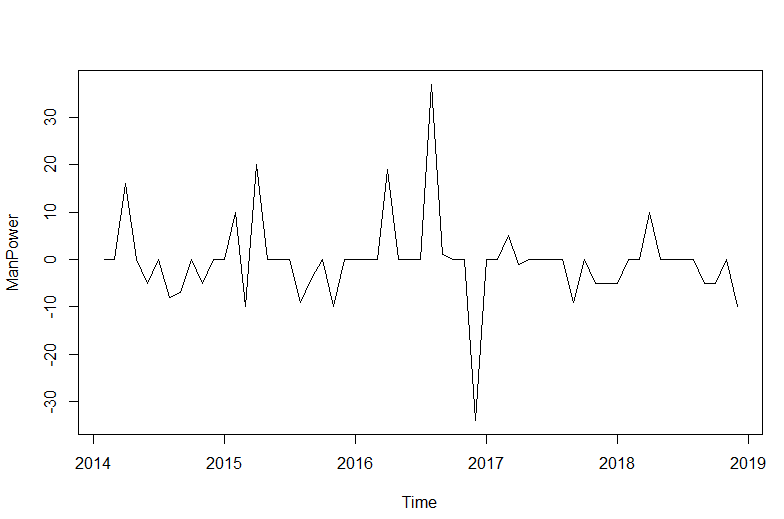


**#As P-value > 0.05, we fail to reject the null hypothesis. Thus, this time series is not stationary.**

**#Differencing as original time series is not stationary**

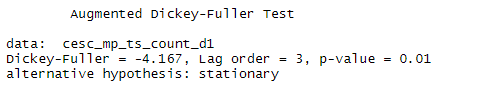
cesc\_mp\_ts\_count\_d1 = diff(cesc\_mp\_ts, differences = 1)

plot(cesc\_mp\_ts\_count\_d1)



**#ADF test for stationarity after first differencing**

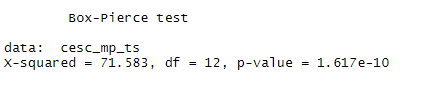
adf.test(cesc\_mp\_ts\_count\_d1, alternative = "stationary")



**#As p-value < 0.05, thus we reject the null hypothesis. Hence, first differencing time series is stationary. Hence, d = 1 for ARIMA model**

**#White noise test for original time series**

Box.test(cesc\_mp\_ts, lag = 12, type = c("Box-Pierce", "Ljung-Box"), fitdf = 0)



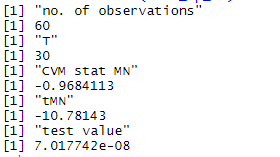
**#As P-value is very small and less than 0.05, we reject the null hypothesis (Series is white noise). Thus, original time series is no white noise.**

**#Another method to test white noise**

install.packages("normwhn.test")

library(normwhn.test)

whitenoise.test(cesc\_mp\_ts)

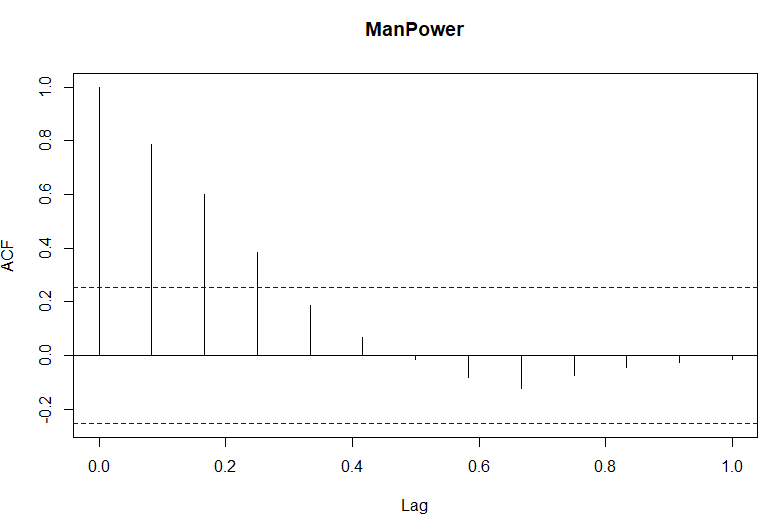


**#As test value is very less than 0.05, hence we again reject the null hypothesis. Series has no white noise.**

**#Plotting ACF, PACF and IACF to check seasonality, trend in original time series and to understand the order of ARIMA model (by checking significant peaks)**

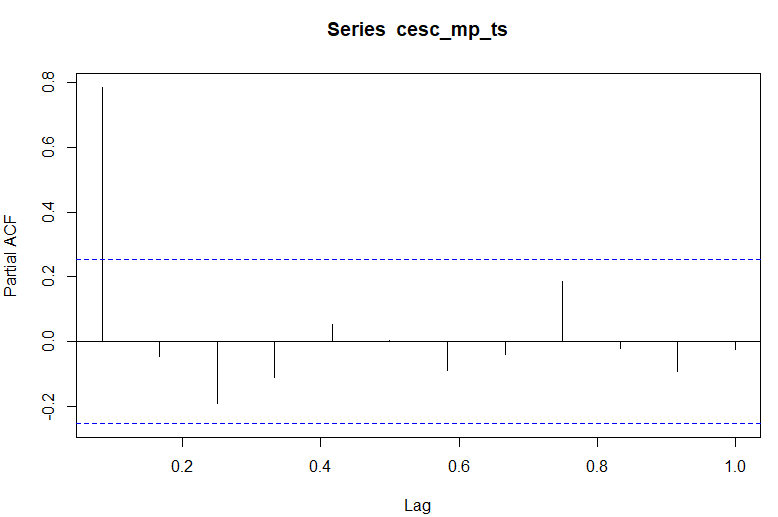
**#ACF and PACF on original time series (before first differencing and Lag = 12)**

acf(cesc\_mp\_ts,lag.max = 12,type = c("correlation", "covariance", "partial"), plot = TRUE, na.action = na.contiguous, demean = TRUE)



**#As we can see from ACF plot that 4 peaks are significant. Thus, y(t) for Manpower forecast depends on y(t-1), y(t-2) and y(t-3) values and q for MA(q) model <= 3**

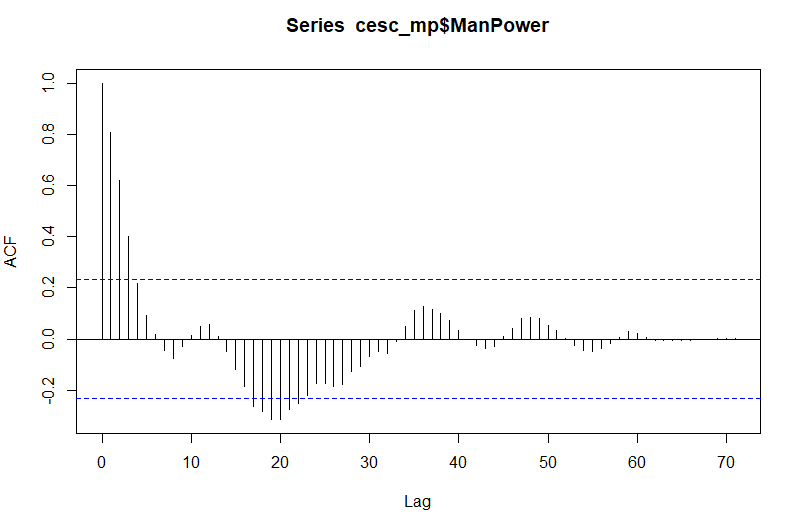
pacf(cesc\_mp\_ts, lag.max = 12)



**# We can see that only first peak is significant. Thus, AR(p), p<=1**

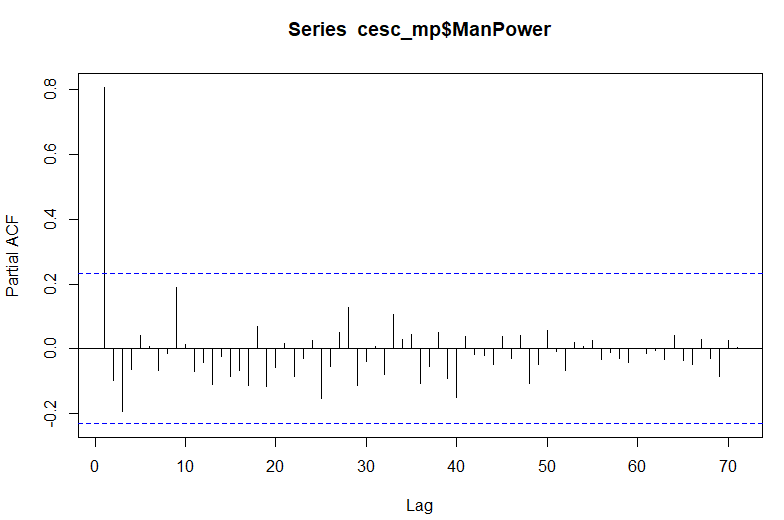
**#ACF and PACF on original time series (before first differencing and Lag = 100)**

acf(cesc\_mp$ManPower,lag.max = 100,type = c("correlation", "covariance", "partial"), plot = TRUE, na.action = na.contiguous, demean = TRUE)



**#ACF plot exponentially decays like in AR (1) model.**

pacf(cesc\_mp$ManPower, lag.max = 100)

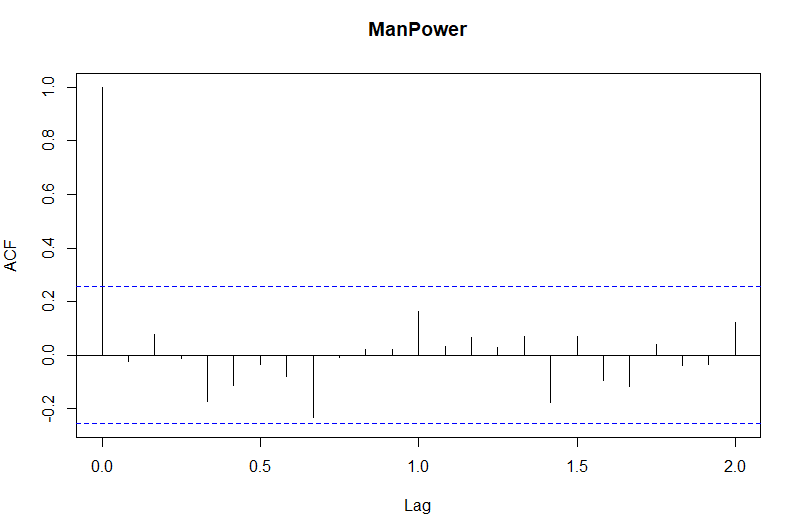


**#PACF plot dies out after Lag = 1 like in AR (1) model.**

**#ACF and PACF for first differencing**

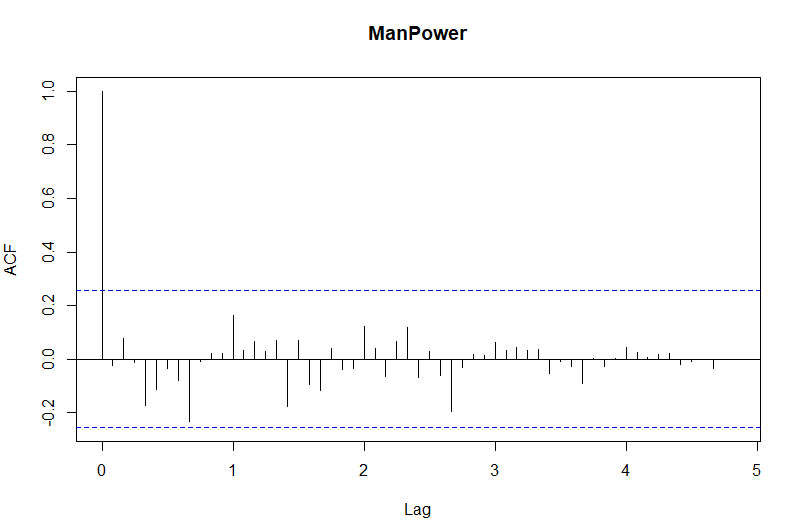
**#ACF with Lag = 24**

acf(cesc\_mp\_ts\_count\_d1,lag.max = 24,type = c("correlation", "covariance", "partial"), plot = TRUE, na.action = na.contiguous, demean = TRUE)



**#ACF with Lag = 100**

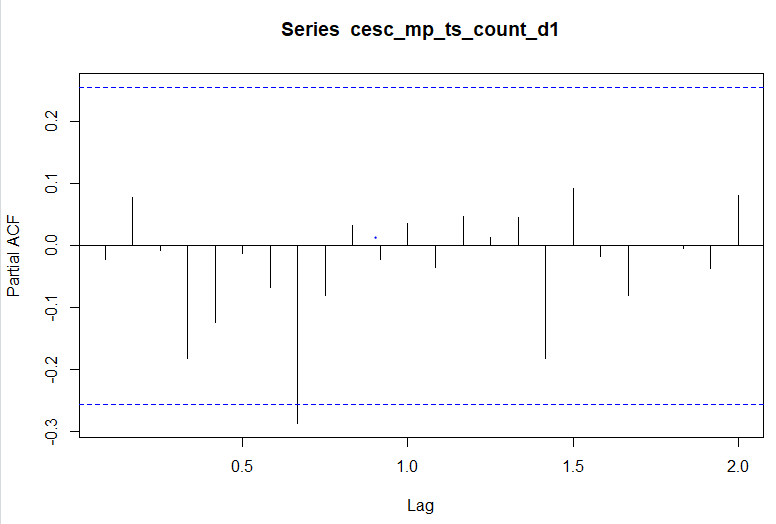
acf(cesc\_mp\_ts\_count\_d1,lag.max = 100,type = c("correlation", "covariance", "partial"), plot = TRUE, na.action = na.contiguous, demean = TRUE)



**#None of the peaks are significant**

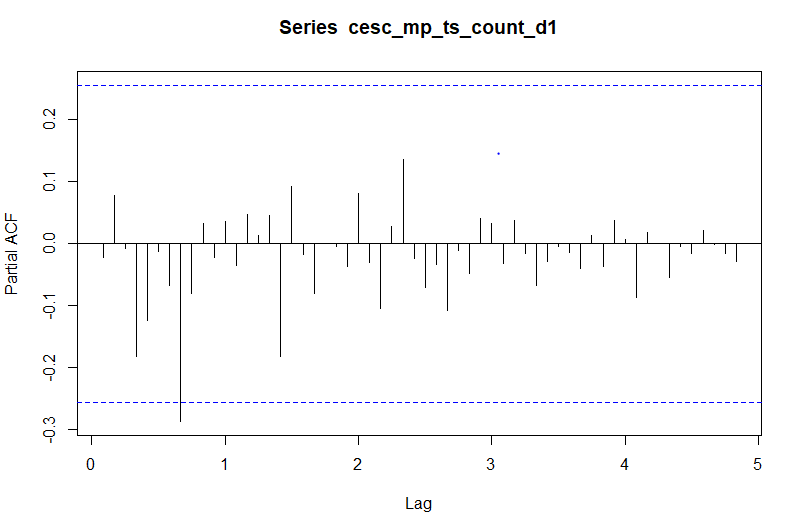
**#PACF with Lag = 24**

pacf(cesc\_mp\_ts\_count\_d1, lag.max = 24)



**#PACF with Lag = 100**

pacf(cesc\_mp\_ts\_count\_d1, lag.max = 100)



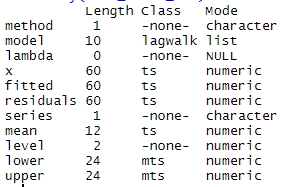
**#On observing PACF plot we can say 8th lag is significant for first differencing**

**#Creating models for Univariate Time Series Analysis**

**#Model using naive method**

cesc\_naive\_mod = naive(cesc\_mp\_ts,h=12)

summary(cesc\_naive\_mod)



**#Comments on predicted values: We can see Naive method predicts the same value for the entire #forecasting horizon.**

**# Checking MAPE function on test data**

cesc\_mp\_test$naive = 25

mape(cesc\_mp\_test$ManPower, cesc\_mp\_test$naive)

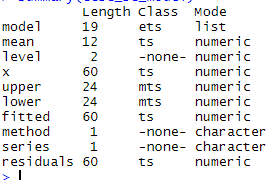


**#MAPE error on test data comes out to be 38 % approximately.**

**#Model using Simple Exponential Smoothing** - extension of the naive method, wherein the forecasts are produced using weighted averages of past observations, with the weights decaying exponentially as the observations get older.

cesc\_se\_model = ses(cesc\_mp\_ts,h=12)

summary(cesc\_se\_model)



**#The output above shows that the simple exponential smoothing has the same value for all the forecasts. Because the alpha value is close to 1, the forecasts are closer to the most recent observations.**

**#Evaluate model performance on test data**

cesc\_se\_model\_pred=as.data.frame(cesc\_se\_model)

cesc\_mp\_test$simpleexp = cesc\_se\_model\_pred$`Point Forecast`

mape(cesc\_mp\_test$ManPower, cesc\_mp\_test$simpleexp)



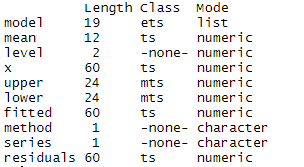
**#MAPE error on test data is 38 % approximately**

**#Holt's Trend Method** - extension of Simple Exponential Smoothing method, which considers Trend component in the Time series while generating forecasts

#This method involves 2 equations - One for Level, another for Trend component

cesc\_holt\_model = holt(cesc\_mp\_ts,h=12)

summary(cesc\_holt\_model)



**#Evaluate model performance on test data**

cesc\_holt\_model\_pred=as.data.frame(cesc\_holt\_model)

cesc\_mp\_test$holt=cesc\_holt\_model\_pred$`Point Forecast`

mape(cesc\_mp\_test$ManPower, cesc\_mp\_test$holt)



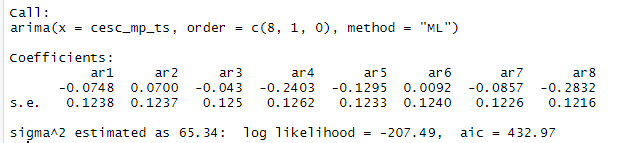
**#MAPE error for this model is 41%, which is worser than previous models.**

**#ARIMA**

**#Fitting ARIMA model of order (8,1,0)**

cesc\_fitARIMA1 = arima(cesc\_mp\_ts, order=c(8,1,0),method="ML")

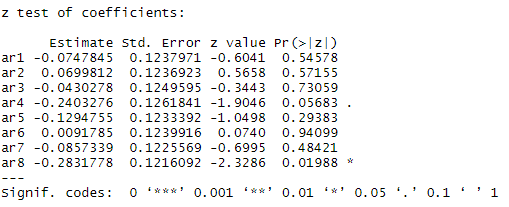
print(cesc\_fitARIMA1)



**#ARIMA (8,1,0) performance metrics**

library(lmtest)

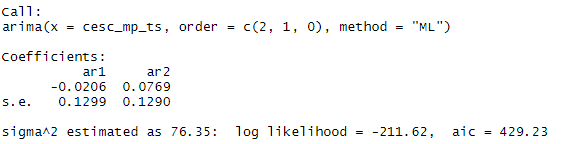
coeftest(cesc\_fitARIMA1)



**#Fit ARIMA (2,1,0) on training set**

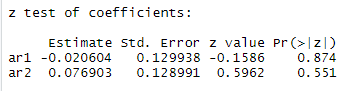
cesc\_fitARIMA2 = arima(cesc\_mp\_ts, order=c(2,1,0),method="ML")

print(cesc\_fitARIMA2)



**#ARIMA (2,1,0) performance metrics**

coeftest(cesc\_fitARIMA2)



**#ARIMA (2,1,0) performance metrics**

library(fpp)

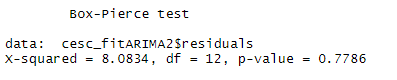
accuracy(cesc\_fitARIMA2)



**#As MASE is approximately 1, model’s accuracy is good.**

**#White Noise test for residuals**

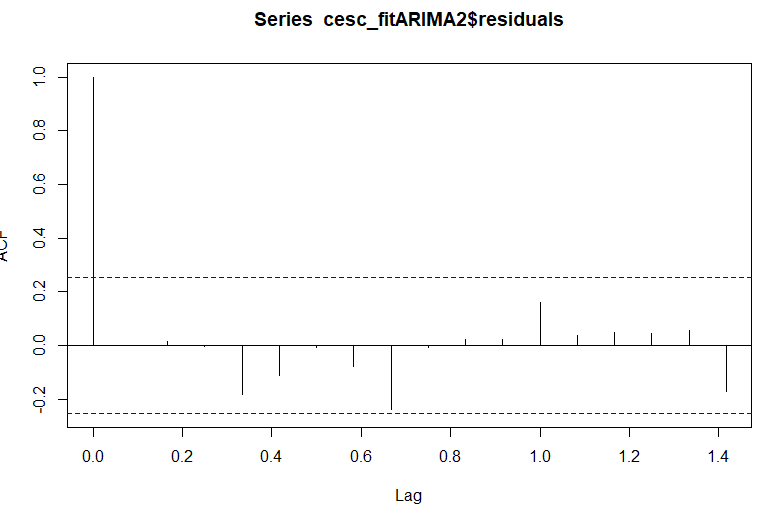
Box.test(cesc\_fitARIMA2$residuals, lag = 12, type = c("Box-Pierce", "Ljung-Box"), fitdf = 0)



**#As P-value > 0.05 , we fail to reject the null hypothesis. Thus, residuals resemble white noise.**

**#ACF of residuals**

acf(cesc\_fitARIMA2$residuals)

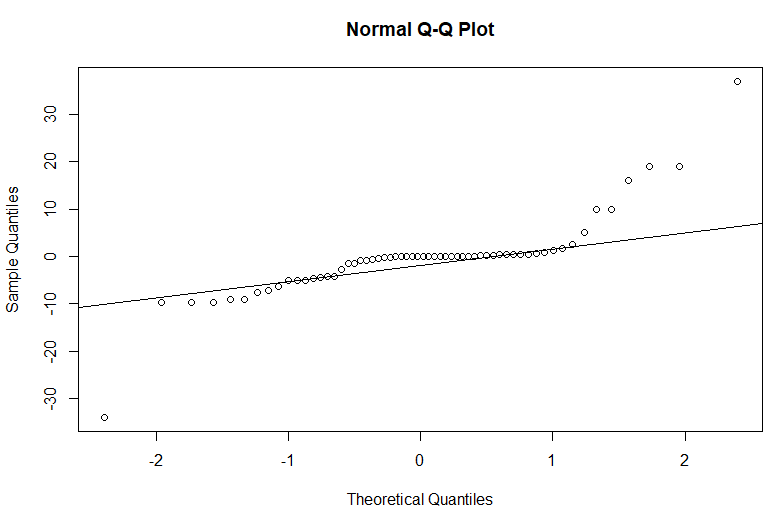


**#ACF has no significant correlations.**

**#Plotting residuals**

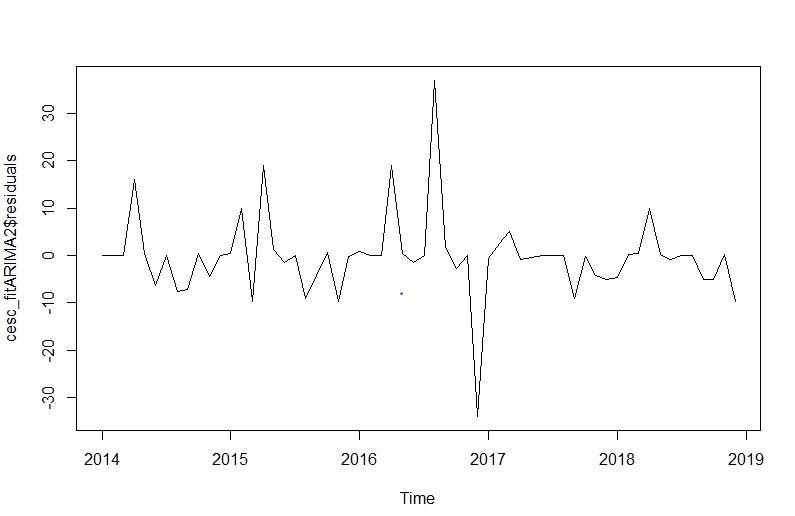
qqnorm(cesc\_fitARIMA2$residuals)

qqline(cesc\_fitARIMA2$residuals)



**#From the results, it looks like the standardized residuals are close to following a normal distribution #except few outliers, so there is no concern about satisfying the normality assumption in the model.**

plot(cesc\_fitARIMA2$residuals)



**#Also, residuals have mean approximately equal to zero and constant variance except between 2016 #and 2017.**

**#Predicting using ARIMA (2,1,0)**

cesc\_arima\_forecast2= predict(cesc\_fitARIMA2,12)

cesc\_arima\_forecast2$pred

mape(cesc\_mp\_test$ManPower, cesc\_arima\_forecast2$pred)



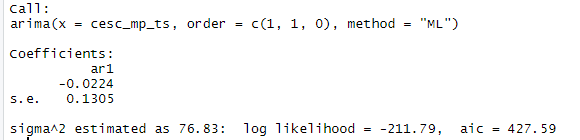
**#MAPE error for prediction is 39%, which is very bad prediction accuracy. This suggests data got overfitted in train dataset.**

**--------------\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*--------------**

**#Fitting ARIMA model of order (1,1,0)**

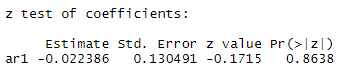
cesc\_fitARIMA3 = arima(cesc\_mp\_ts, order=c(1,1,0),method="ML")

print(cesc\_fitARIMA3)



**#ARIMA (1,1,0) performance metrics**

coeftest(cesc\_fitARIMA3)



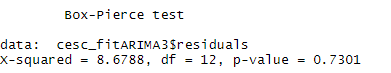
accuracy(cesc\_fitARIMA3)



**#MAPE error is 9%, which is much better than the rest of the models.**

**#White Noise test for residuals**

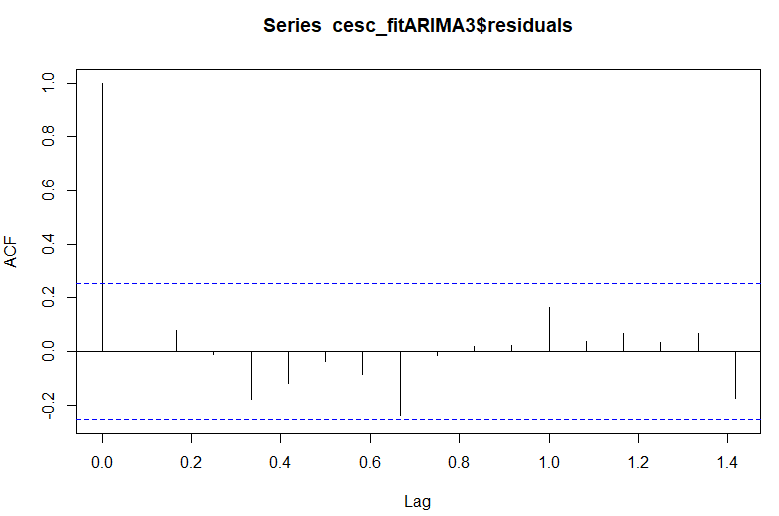
Box.test(cesc\_fitARIMA3$residuals, lag = 12, type = c("Box-Pierce", "Ljung-Box"), fitdf = 0)



**#As P-value > 0.05 , we fail to reject the null hypothesis. Thus, residuals resemble white noise.**

**#ACF of residuals**

acf(cesc\_fitARIMA3$residuals)

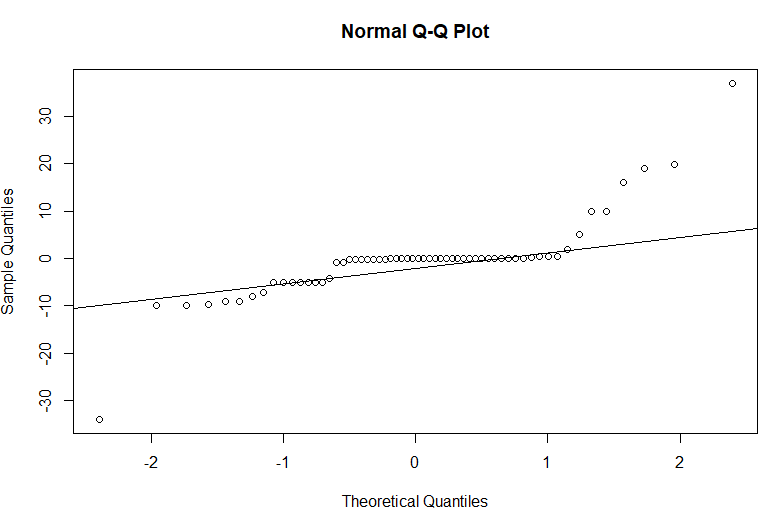


**#ACF has no significant correlations.**

**#Plotting residuals**

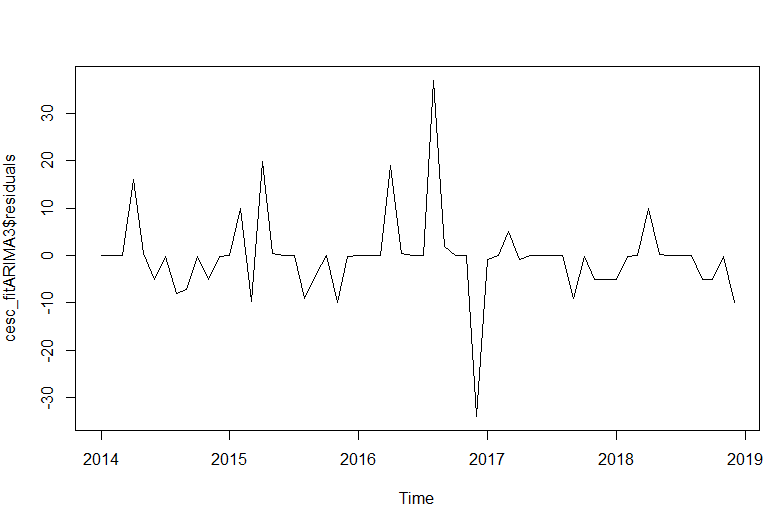
qqnorm(cesc\_fitARIMA3$residuals)

qqline(cesc\_fitARIMA3$residuals)



**#From the results, it looks like the standardized residuals are close to following a normal distribution #except few outliers, so there is no concern about satisfying the normality assumption in the model.**

plot(cesc\_fitARIMA3$residuals)



**#Also, residuals have mean approximately equal to zero and constant variance except one spike.**

**#Predicting using ARIMA (1,1,0)**

cesc\_arima\_forecast3= predict(cesc\_fitARIMA3,12)

cesc\_arima\_forecast3$pred

mape(cesc\_mp\_test$ManPower, cesc\_arima\_forecast3$pred)



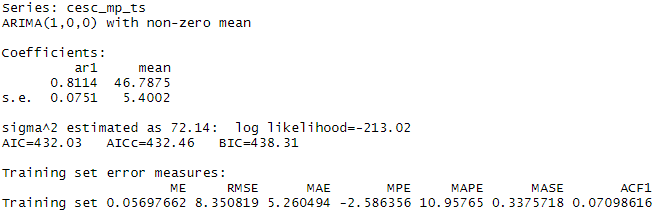
**#MAPE error for prediction is 38%, which is very bad prediction accuracy. This suggests data got overfitted in train dataset.**

**--------------\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*---------**

**#ARIMA model (with auto.arima)**

cesc\_arima\_model = auto.arima(cesc\_mp\_ts)

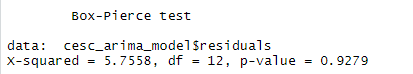
summary(cesc\_arima\_model)



**#MAPE error is 11%, which is much better than the rest of the models but more than ARIMA (1,1,0).**

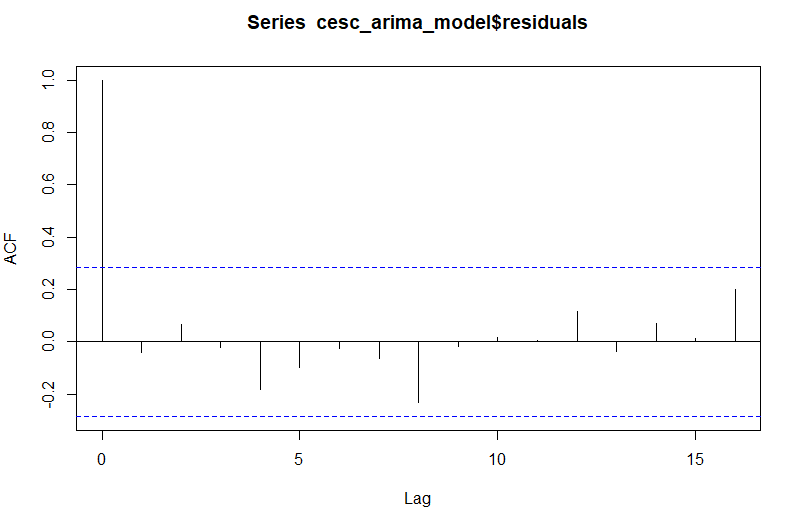
**#White Noise test for residuals**

Box.test(cesc\_arima\_model$residuals, lag = 12, type = c("Box-Pierce", "Ljung-Box"), fitdf = 0)



**#As P-value > 0.05 , we fail to reject the null hypothesis. Thus, residuals resemble white noise.**

**#ACF of residuals**

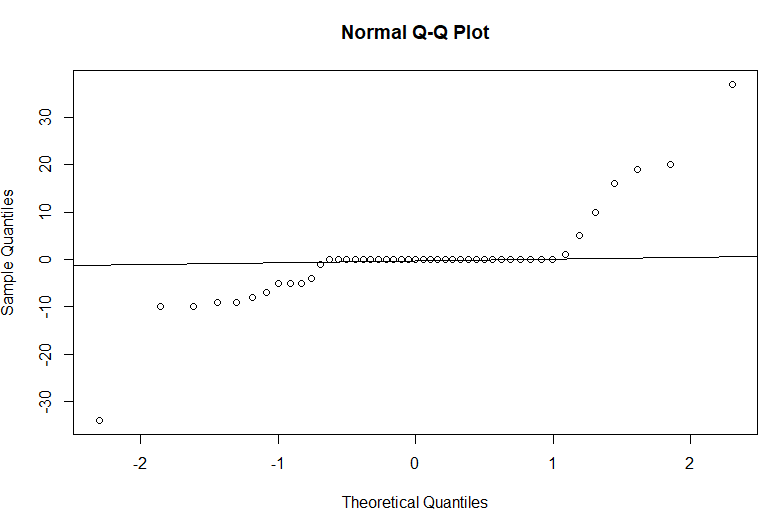
acf(cesc\_arima\_model$residuals)

**#ACF has no significant correlations.**

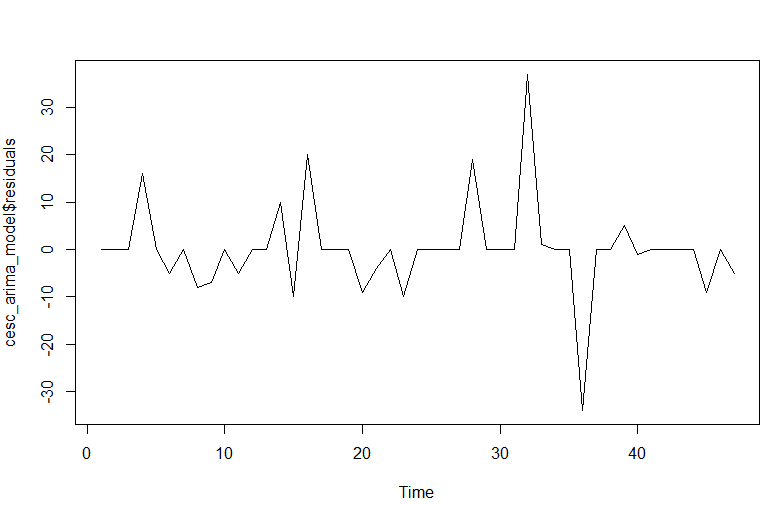
**#Plotting residuals**

qqnorm(cesc\_fitARIMA2$residuals)

qqline(cesc\_fitARIMA2$residuals)



**#From the results, it looks like the standardized residuals are close to following a normal distribution #except few outliers, so there is no concern about satisfying the normality assumption in the model. #But there is no trend in the residuals.**



**#Also, residuals have mean approximately equal to zero and constant variance except one spike.**

**#Predicting using ARIMA (1,0,0) – for Test set and year 2020**

cesc\_autoarima\_forecast= predict(cesc\_arima\_model,24)

cesc\_autoarima\_forecast$pred

mape(cesc\_mp\_test$ManPower, cesc\_autoarima\_forecast$pred)



**#MAPE error for prediction is 12% when predicting for a year but the error rises to 14% when predicting for 2 years. However, this is far better than other ARIMA models. So, we choose ARIMA(1,0,0) for prediction.**

**#Creating prediction dataframe using auto ARIMA (1,0,0) model as MAPE on training set is 11% and MAPE for prediction accuracy is 12%, which is better than other models.**

cesc\_mp\_final = rbind(cesc\_mp[61:72,], c("2020-01-01",".","New"), c("2020-02-01",".","New"), c("2020-03-01",".","New"), c("2020-04-01",".","New"), c("2020-05-01",".","New"), c("2020-06-01",".","New"), c("2020-07-01",".","New"), c("2020-08-01",".","New"), c("2020-09-01",".","New"), c("2020-10-01",".","New"), c("2020-11-01",".","New"), c("2020-12-01",".","New"))

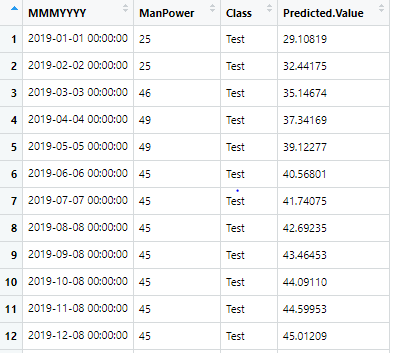
cesc\_mp\_final=as.data.frame(cesc\_mp\_final)

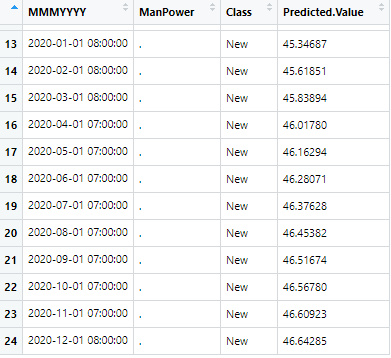
autoarima\_fr=data.frame("Predicted Value" =cesc\_autoarima\_forecast$pred)

cesc\_mp\_final = cbind(cesc\_mp\_final, 'Forecast' = autoarima\_fr)

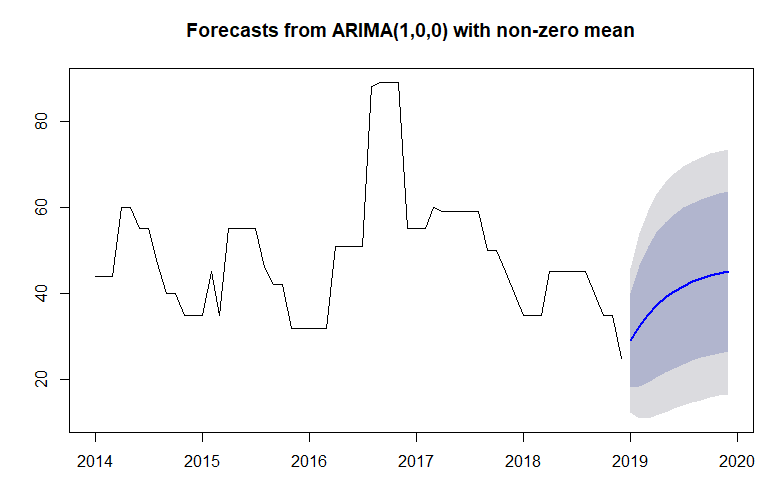
View(cesc\_mp\_final)

**#Manpower Predictions for test set (year 2019) and Year 2020**





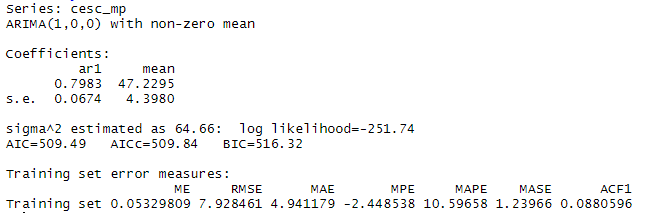
**#Manpower Forecast plot**



**#Fitting the model on whole dataset**

cesc\_arima\_model\_full = auto.arima(cesc\_mp)

summary(cesc\_arima\_model\_full)



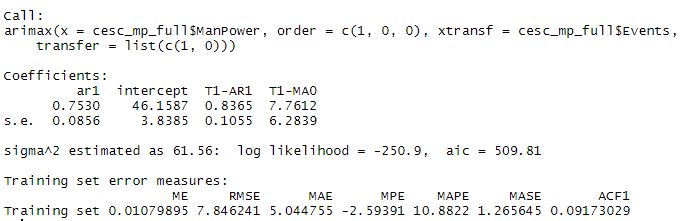
cesc\_autoarima\_forecast\_full= predict(cesc\_arima\_model\_full,12)

cesc\_autoarima\_forecast\_full$pred

mape(cesc\_mp\_test$ManPower, cesc\_autoarima\_forecast\_full$pred)

**#We are getting same MAPE error and better RMSE. So, we will go with ARIMA(1,0,0) model for univariate time series forecasting.**

**#Applying intervention analysis -December 2016: Point transfer function and August 2016: Step function**



**---------------\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*--------**

**#Creating model using FACEBOOK PROPHET**

install.packages("prophet")

library(prophet)

install.packages("data.table")

library(data.table)

install.packages("dplyr")

library(dplyr)

install.packages("ggplot2")

library("ggplot2")

**#Building the model on whole dataset and replacing the names of columns. “ ds” = Date column and “y” = Manpower (target variable)**

cesc\_mp\_fp=cesc\_mp

cesc\_mp\_fp = cesc\_mp\_fp[-3]

names(cesc\_mp\_fp) = c("ds", "y")

View(cesc\_mp\_fp)

**#Model Building after fixing maximum (120 agents) and minimum (10 agents) limits, reducing seasonality and adjusting trend using changepoints**

cesc\_mp\_fp$cap = 120

cesc\_mp\_fp$floor = 10

m4 = prophet(cesc\_mp\_fp, changepoint.prior.scale = 0.4, yearly.seasonality = 5)

**# Visualize forecast**

cesc\_prophet\_future3 = make\_future\_dataframe(m4, periods=12, freq = "month")

cesc\_prophet\_future3$cap = 120

cesc\_prophet\_future3$floor = 10

cesc\_prophet\_forecast3 = predict(m4, cesc\_prophet\_future3)

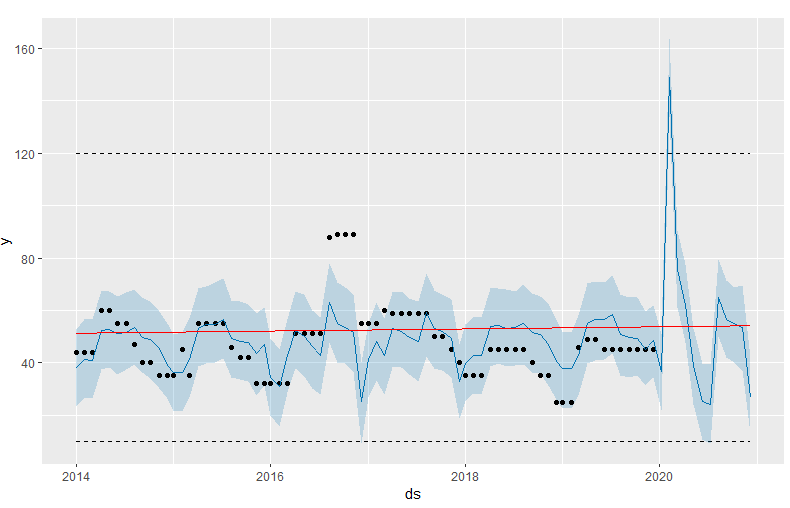
cesc\_mp\_prophet\_pred3 = cesc\_prophet\_forecast3[c('ds','yhat','yhat\_lower','yhat\_upper')]

View(cesc\_mp\_prophet\_pred3)

**#Observing changepoints [ Blue line: predicted values, Black dots: actual values , Red line: Changepoints (sudden events or changes) ]**

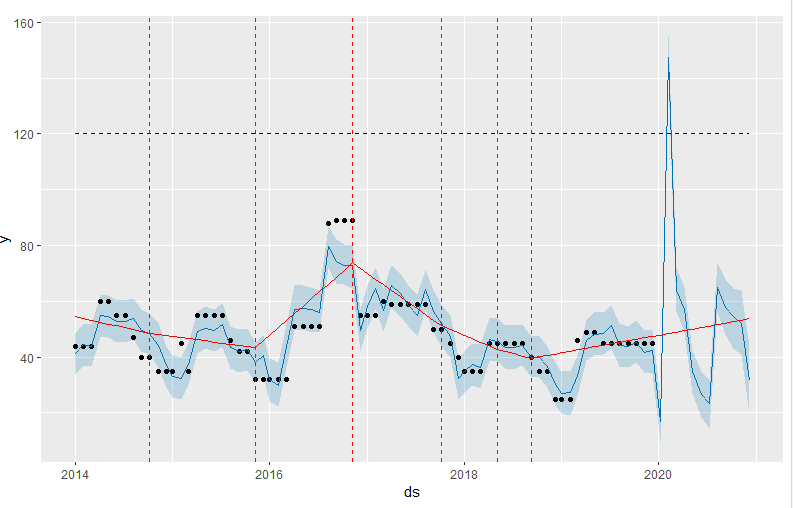
plot(m4, cesc\_prophet\_forecast3) + add\_changepoints\_to\_plot(m4)

**#With default changepoints**

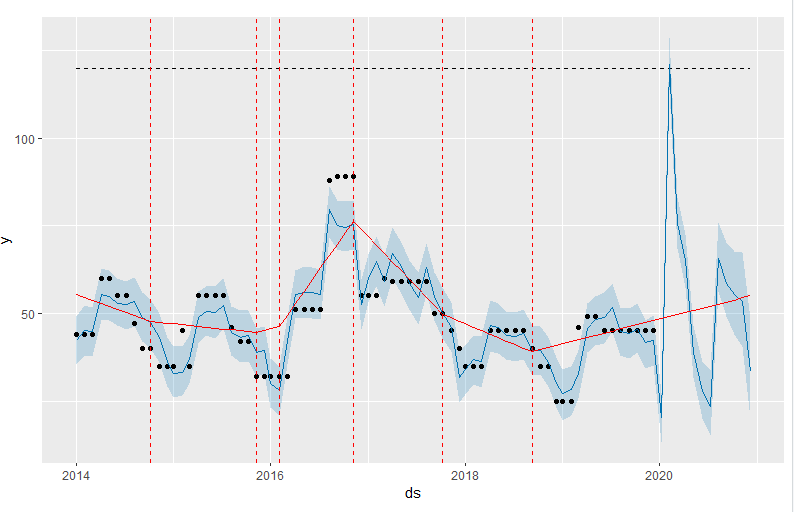


**#After adjusting only trend flexibility**

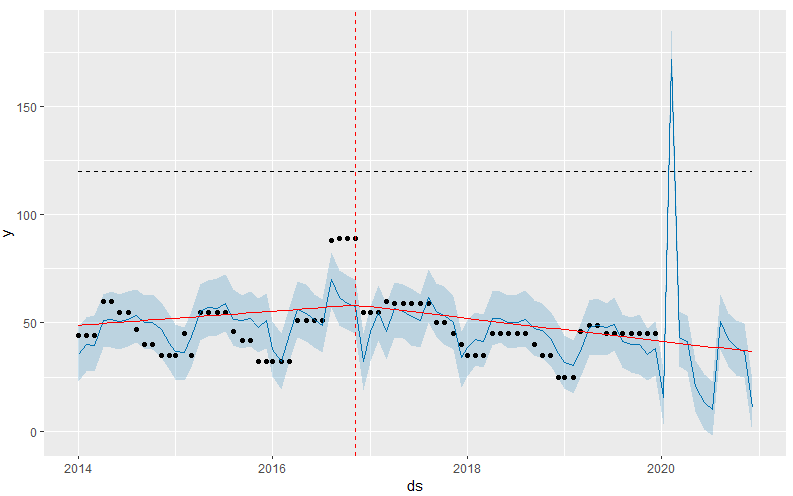
**#When changepoint.prior.scale = 0.4**



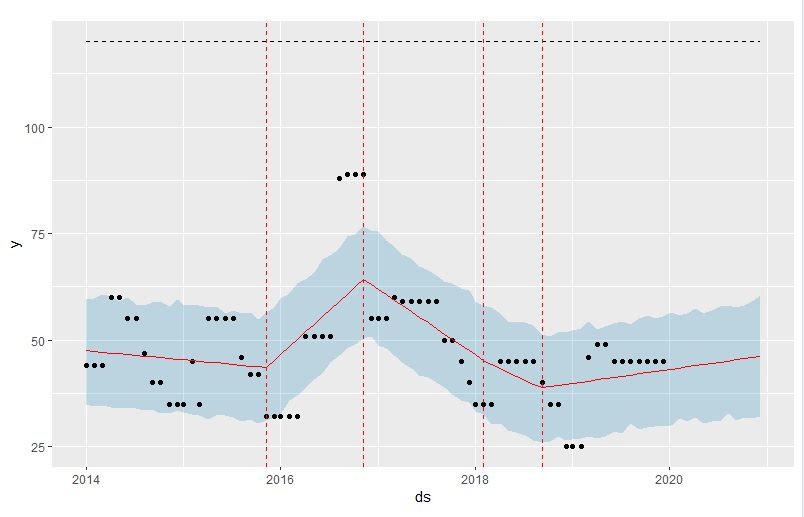
**#When changepoint.prior.scale = 0.5**



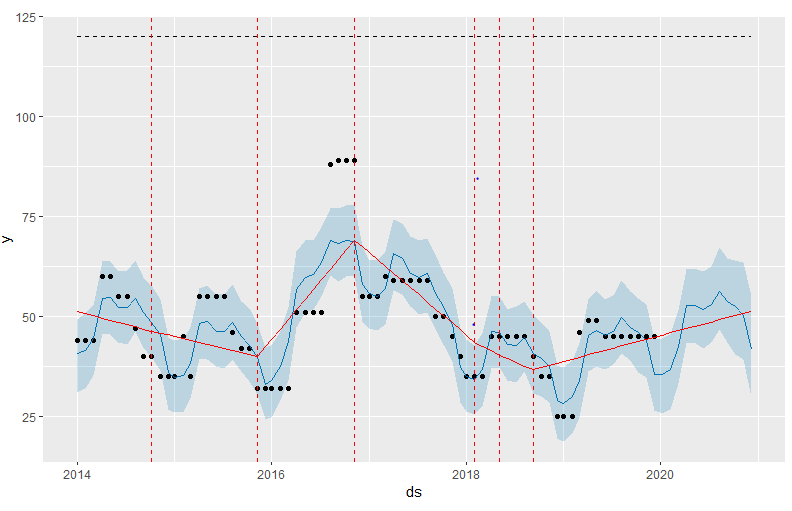
**#When changepoint.prior.scale = 0.01**



**#After adjusting trend flexibility and NO seasonality component**

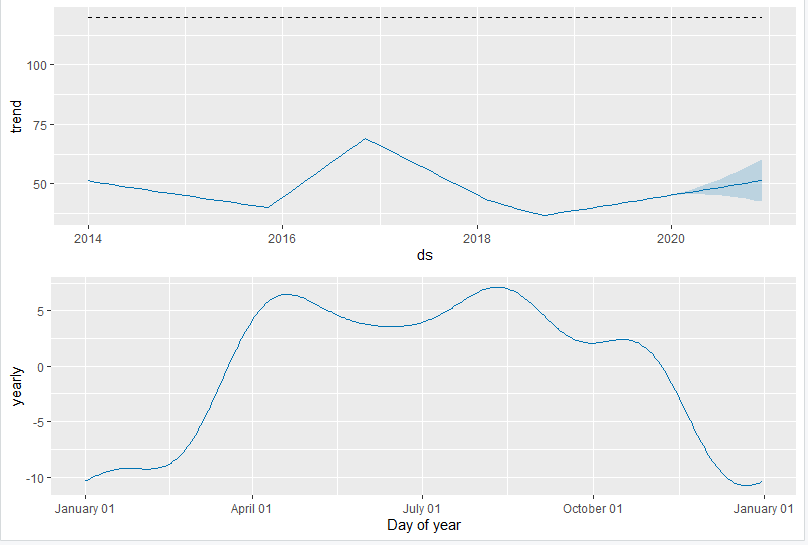


**#After adjusting trend flexibility and reducing seasonality component**



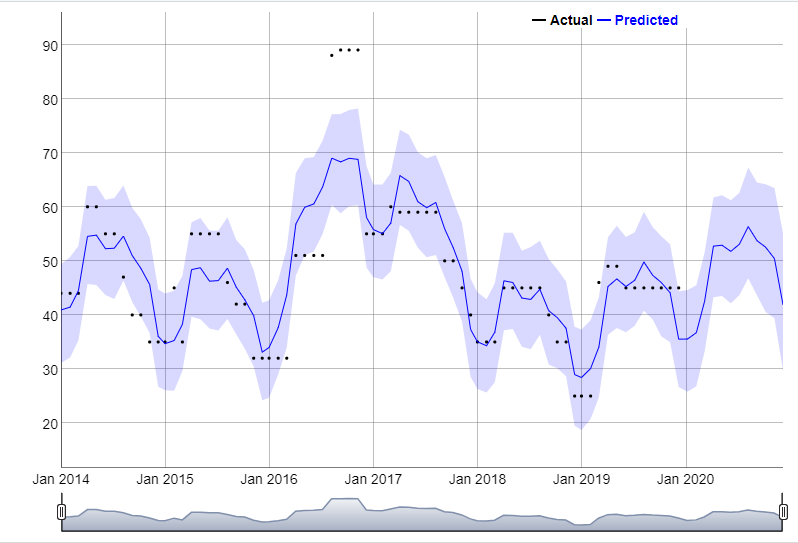
**#Plotting trend, weekly seasonality and yearly seasonality**

prophet\_plot\_components(m4, cesc\_prophet\_forecast3)

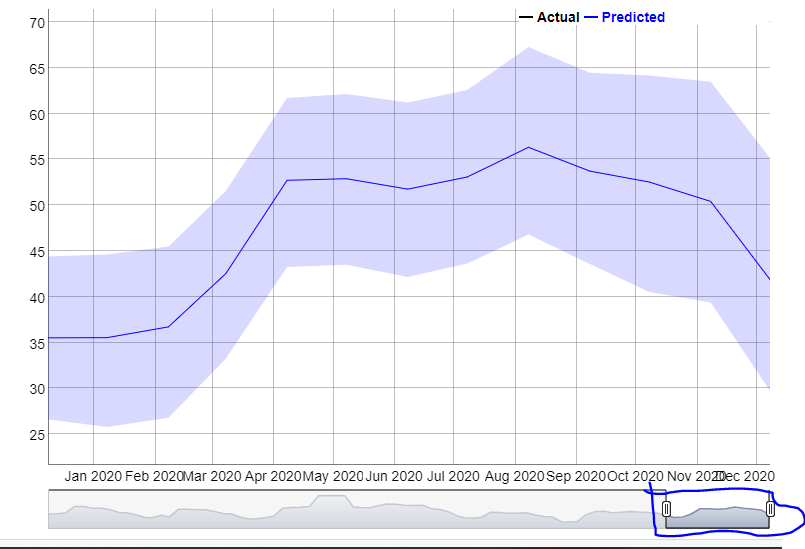


**#Interactive plot**

dyplot.prophet(m4, cesc\_prophet\_forecast3)



**#On expanding the prediction range of interactive plot:**



**#Model checking for Prophet**

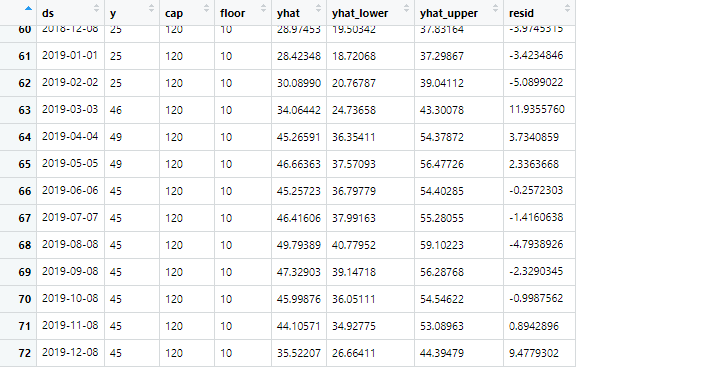
**#Merge original and predicted datasets to get the residuals**

merge\_fp = merge(x=cesc\_mp\_fp,y=cesc\_mp\_prophet\_pred3,by="ds")

View(merge\_fp)

**#Residuals**

merge\_fp$resid=merge\_fp$y - merge\_fp$yhat



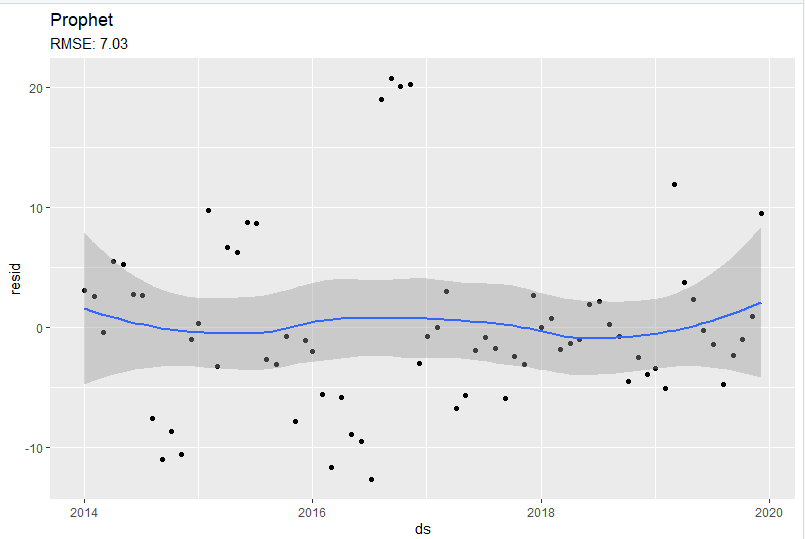
**#Plot the residuals – evaluating forecasting accuracy**

ggplot(merge\_fp,aes(ds,resid)) +

geom\_point() + geom\_smooth() +

ggtitle("Prophet",

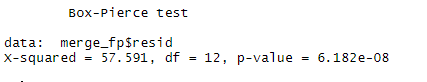
subtitle = paste0("RMSE: ",round(sqrt(mean(merge\_fp$resid^2)),2)))



Model has a good RMSE. We will further verify this using crossvalidation.

**#Whitenoise test on residuals**

Box.test(merge\_fp$resid, lag = 12, type = c("Box-Pierce", "Ljung-Box"), fitdf = 0)

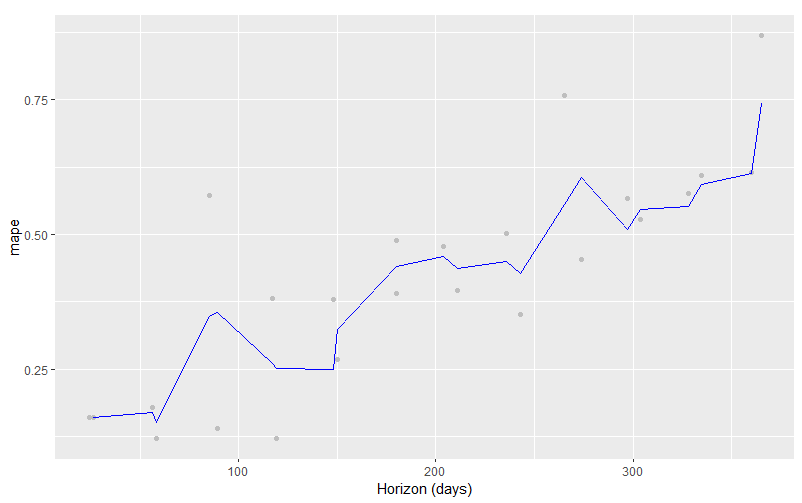


**#As P-value is < 0.05 , we reject the null hypothesis. Thus, residuals resemble white noise. SO model can be improved.**

**#Checking performance using Cross validation**

cesc\_fp\_cv = cross\_validation(m4, initial = 1525, period = 180, horizon = 365, units = 'days')

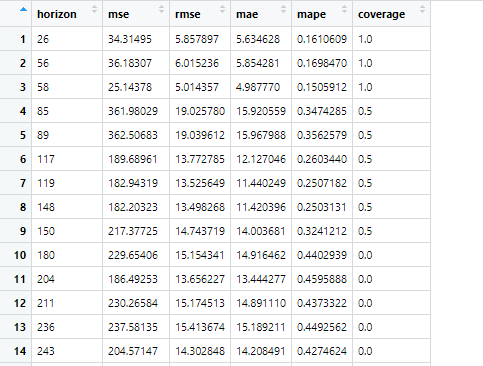
plot\_cross\_validation\_metric(cesc\_fp\_cv, metric = 'mape')

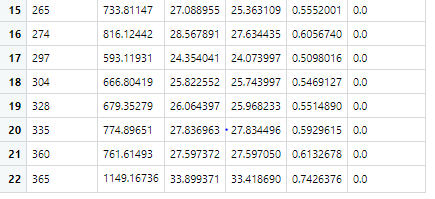


View(cesc\_fp\_cv)

cesc\_fp\_perf = performance\_metrics(cesc\_fp\_cv)

View(cesc\_fp\_perf)



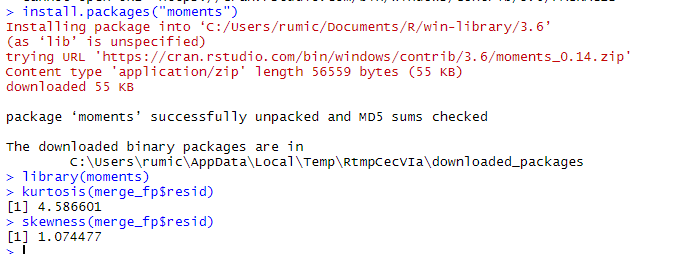


**#We can see MAPE is 1% for predicting 26 days and increases to 7.5% when we predict past one year. This percentage of error #is acceptable and highly accurate.**

**#Mean and median of residuals**

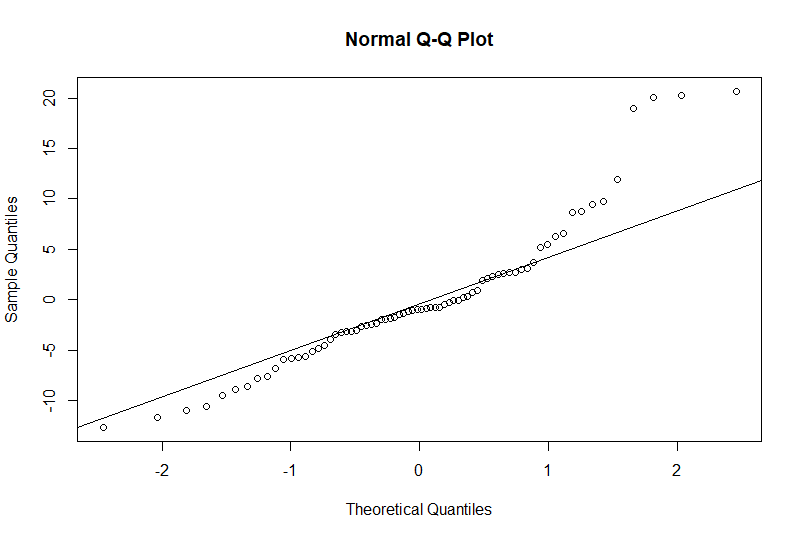
****

**#Kurtosis and Skewness of residuals**

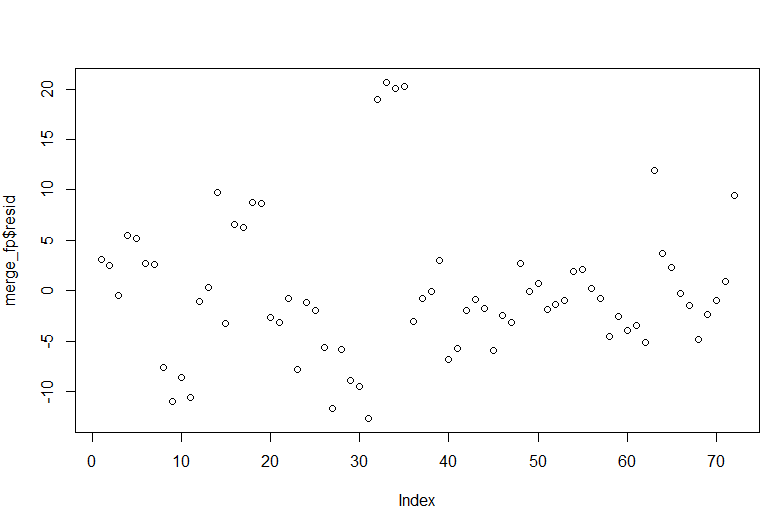


**#Plot for residuals – there seems to be a little correlation between residuals**

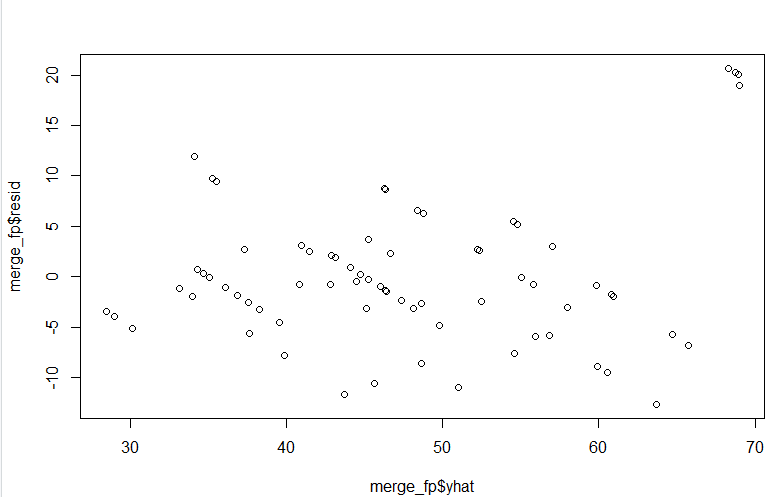
**#QQ-Plot of residuals**



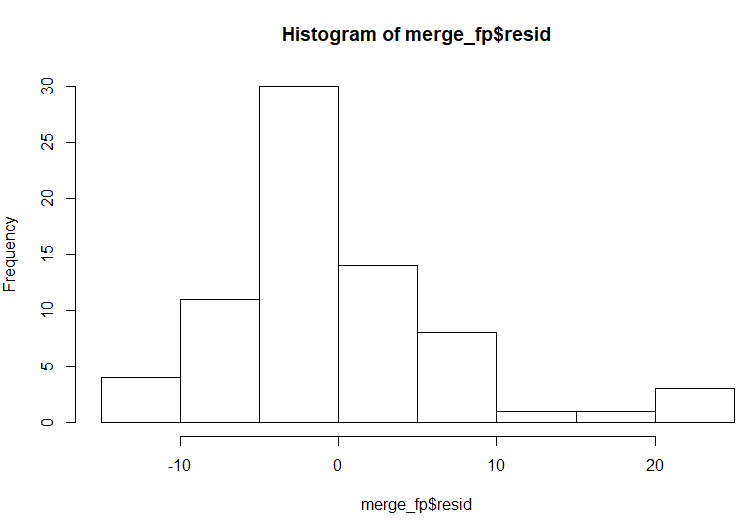
**#From the results, it looks like the standardized residuals are close to following a normal distribution #except few outliers, so there is no concern about satisfying the normality assumption in the model.**



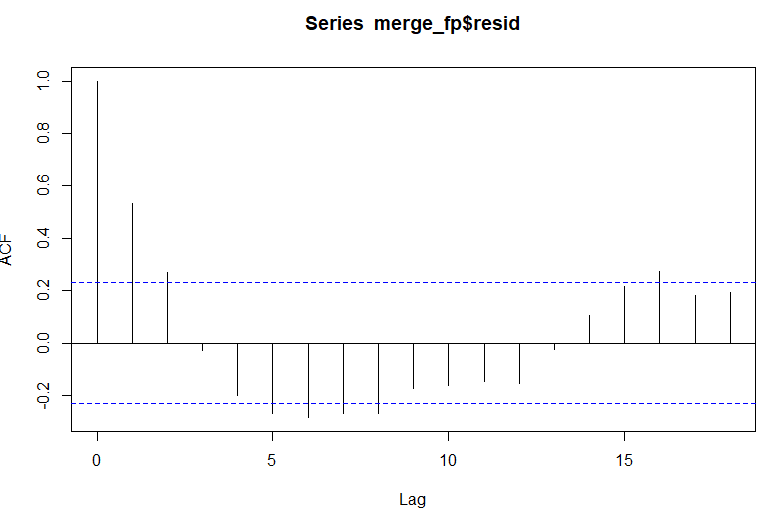
**#Plot – Residuals vs Predicted values**



**#Histogram of residuals: Mean of the plot is almost zero but the plot is a bit skewed towards left.**



**#ACF of residuals:**



#**As we can see residuals are somewhat correlated, the model predictions are good for short term forecast and not recommended for long term.**